UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
ESPECIALIZAÇÃO SEMIPRESENCIAL EM CONFIGURAÇÃO E GERENCIAMENTO DE SERVIDORES E EQUIPAMENTOS DE REDES

MARCELO VEIGA PEREIRA

IMPLEMENTANDO SEGURANÇA NO NÍVEL DE ACESSO UTILIZANDO SERVIDOR RADIUS

MONOGRAFIA

CURITIBA
2011

MARCELO VEIGA PEREIRA

IMPLEMENTANDO SEGURANÇA NO NÍVEL DE ACESSO UTILIZANDO SERVIDOR RADIUS

Monografia apresentada como requisito para obtenção do título de Especialista Configuração e Gerenciamento de Servidores e Equipamentos de Redes pela Universidade Tecnológica Federal do Paraná, UTFPR.

Orientador: Prof. Dr. Augusto Foronda

CURITIBA
2011

RESUMO

PEREIRA, Marcelo V. *Implementando Segurança no Nível de Acesso Utilizando Servidor RADIUS*. 2011. Monografia (Especialização em Configuração e Gerenciamento de Servidores e Equipamentos de Redes), Universidade Tecnológica Federal do Paraná, UTFPR, Curitiba, 2011.

Este trabalho tem como tema central apresentar as facilidades disponibilizadas pelo serviço de autenticação disponíveis em um Servidor RADIUS em conjunto com banco de dados MySQL. Tendo como base a necessidade de segurança dos pontos de acesso de rede sem fio, a mobilidade dos usuários e a facilidade de utilizar uma base de dados centralizada de autenticação.

Palavras-chaves: RADIUS. Segurança. Redes. Acesso.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

1.1 TEMA

Com a popularização dos computadores e o acesso fácil a informações, graças a internet, os profissionais de informática precisaram desenvolver novos métodos para dar segurança aos dados que trafegam por seus sistemas.

Com a facilidade de acesso a informação surgiram os primeiros Hackers, que utilizavam seus conhecimentos para pregar peças através da grande rede ou às vezes roubar informações importantes. Esse ponto negativo levou a necessidade da formação de profissionais especializados ao combate desse tipo de ataque.

Desse ponto em diante o termo Segurança da Informação tornou-se popular.

Havia a necessidade de profissionais que tivessem o mesmo conhecimento que seus adversários, os Hackers. Dando origem aos chamados Hackers Éticos que inclusive tem certificações específicas para tal através de organizações como The International Council of E-Commerce Consultants (EC-Council).

O objetivo de toda essa segurança é proteger informações de pessoas que não tem autorização a obtê-las. Dificultando seu acesso a elas fortificando barreiras com firewalls e DMZs.

Além dos cuidados com a segurança contra invasões também é preciso o cuidado com o acesso a instalações físicas.A organização e projeto as instalações físicas também são importante para a segurança da informação porque uma vez tendo acesso a essas instalações não é preciso já não mais é preciso atravessar outras barreiras como firewall. Por isso é preciso que a instalações sejam estruturadas seguindo normas EIA/TIA não só para segurança física do pessoal envolvido tanto quanto para a segurança ao acesso as instalações.

Pode-se citar como exemplo o acesso do técnico da operadora de serviços de longa distância ao ponto de demarcação. É preciso que os demais equipamentos não estejam acessíveis para garantir a segurança.

Uma das formas de controlar esse acesso é através de um servidor autenticação RADIUS (*Remote Authentication Dail In User Service*) e equipamento compatível.

Este trabalho limita-se a implementação de servidor RADIUS para controle de acesso em redes sem fio. Incluindo configurações para autenticação em banco de dados MySQL.

Não serão abordados cenários com redes remotas como: autenticação de cliente em Provedor de Serviços de Internet ou redes com autenticação entre Filiais e seus Escritórios Centrais.

A autenticação entre redes remotas pode ser realizada através de VPN (*Virtual Private Network*) ou outra técnica de tunelamento sobre a qual protocolo RADIUS pode funcionar de modo transparente.

1.2 PROBLEMA E PREMISSAS

Há pouca documentação disponível sobre a configuração de servidores RADIUS. As documentações existentes são extremamente técnicas e voltadas principalmente para os detalhes do protocolo.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Apresentar uma solução prática e simples para o problema de segurança em nível de acesso em redes sem fio que não dificulte a mobilidade de seus usuários.

1.3.2 Objetivos Específicos

- Elencar os aspectos técnicos do protocolo RADIUS que permitam uma posterior implementação deste.
- Apresentar a configuração necessária para o correto funcionamento da autenticação via RADIUS para uma rede sem fio.
- Documentar a configuração de integração de um servidor RADIUS com o banco de dados MySQL..
- Apresentar os benefícios do uso de um servidor RADIUS na segurança e mobilidade dos usuários de uma rede local.

1.4 JUSTIFICATIVA

Em um projeto de rede estruturada há a necessidade de prever o aumento da quantidade de número de usuários. Instala-se mais pontos do que o número de usuários para que seja possível atender novos usuários sem a necessidade de reestruturação.

Se estes pontos adicionais estiverem devidamente habilitados, qualquer usuário que tiver acesso físico ao ponto poderá se conectar imediatamente a rede. Isso permite que usuários se desloquem entre departamentos sem a necessidade de alteração na rede.

Manter todos os pontos habilitados permitem que outras pessoas, mesmo não autorizadas, acessem a rede; desde que tenham acesso físico a um ponto de rede habilitado.

Uma alternativa para o problema do acesso indevido é desabilitar os pontos que não estão sendo utilizados e habilitá-los somente na chegada de um novo usuário. Resolve-se dessa forma o problema dos pontos de acesso vulneráveis mas em contra partida a e realocação de usuários exige que o administrador da rede reabilite o novo ponto e desabilite o ponto que não mais será utilizado. Esse excesso de manobras também exigem a constante atualização da documentação da rede e de como os pontos estão interconectados.

O uso de um servidor RADIUS, em conjunto com equipamentos que suportem esse protocolo, pode resolver o problema de segurança e de mobilidade simultaneamente. A autenticação em nível de usuário permite que todos os pontos estejam fisicamente habilitados mas só estejam ativos para usuários devidamente registados. Dessa forma os usuários podem ser realocados entre departamentos sem a necessidade de reabilitação dos pontos de acesso e o acesso de pessoal não autorizado é impossibilitado pela exigência de informações de autenticação.

1.5 PROCEDIMENTOS METODOLÓGICOS

Pesquisa bibliográfica:

- Serão levantados e documentados dados técnicos sobre a especificação do protocolo RADIUS baseando na RFC 2865.

- Nessa etapa também serão documentados os softwares servidores RADIUS disponíveis e suas características, bem como a escolha de um destes para a realização dos testes.

Laboratório de implementação:

- Nessa etapa será montada uma rede de testes de autenticação utilizando-se um roteador sem fio com suporte ao protocolo RADIUS, como o LinkSys WRT120N, e um computador onde estarão instalados os servidores RADIUS, LDAP e MySQL.

Apresentação das informações de configuração:

- Serão documentados os arquivos, comandos e passos utilizados para a configuração do Servidor e do Roteador sem fio com a opção de autenticação em banco de dados MySQL.

2 REFERENCIAIL TEÓRICO

2.1 IEEE 802.1X

O padrão IEEE 802.1X define um mecanismo de autenticação para equipamentos ou terminais ingressantes em uma rede.

Constituido por três compotentes suplicante, autenticador e um servidor de autenticação.

Suplicante é o equipamento ou terminal de usuário que pretende-se adicionar a rede.

Autenticador é o equipamento de rede (switch ou ponto de acesso sem fio, por exemplo) que intermediário entre o suplicante e o servidor de autenticação.

O Servidor de Autenticação é uma aplicação responsável por responder as requisições de acesso a rede, autorizando ou não o ingresso de suplicantes utilizando uma base de dados de credenciais.

2.2 AAA - *Authentication, Authorization and Accounting*

2.2.1 Autenticação (*Authentication*)

É o processo de se identificar um usuário ou equipamento partindo de uma declaração do requisitante e fazendo a checagem em um banco de informações para constatar a veracidade da declaração.

OS metodos mais comuns de autenticação envolvem usuário e senha outro não tão comum e o uso de certificados digitais.

2.2.2 Autorização (*Authorization*)

É o processo de definir as permissões de acesso de um usuário ou equipamento.

Depois do processo de autenticação é preciso consultar uma segunda base de informações que definem quais os recursos que este usuário tem acesso.

Autorização de acesso envolvem politicas de segurança que devem estar bem documentadas para impedir que usuários acessem informações ou recursos as quais não deveriam ter acesso.

2.2.3 Bilhetagem (*Accounting*)

É o processo de coletar informações a respeito do uso dos recursos utilizados por um usuário para posteriormente sejam emitidas faturas referentes ao tempo de uso. Este recurso é utilizado por ISPs (*Internet Service Providers*) ou Provedores de Serviços de Internet que fornecem serviços de transferência de dados sob demanda para contabilizar o uso do serviço por seus usuários possibilitando cobrança (HASSEL, 2002).

Bilhetagem não se aplica a redes locais porque os serviços disponíveis geralmente são para uso comum ou para pessoal autorizado mas normalmente não estão sujeitos a cobrança.

2.3 Mobilidade (*Roaming*)

Mobilidade é a facilidade que um usuário pode ter de mudar de localidade sem perder o acesso aos recursos da rede. Um usuário poderia se deslocar entre filiais de uma empresa e utilizar as mesmas informações de autenticação em qualquer uma delas. Para isso seria preciso de uma base de autenticação centralizada que poderia funcionar sobre uma VPN.

No caso de ISPs com parceria com empresas de outras de diferentes localidades seus serviços poderiam se expandir por várias cidades.

Pode-se estender o termo a redes locais com usuários sendo realocados entre departamentos.

2.4 *Remote Authentication Dial-In User Service* (RADIUS)

É um protocolo da camada de aplicação do modelo OSI que provê serviços centralizados de Autenticação, Autorização e Bilhetagem de acordo com a definido

pela RFC 2866. O servidor RADIUS utiliza o protocolo UDP da camada de transporte para a troca de mensagens com as aplicações clientes através da porta 1812 para Autenticação e 1813 para Autorização.

2.5 *Extensible Authentication Protocol* (EAP)

EAP (RFC 3748) pode ser descrito como um protocolo com suporte a múltiplos métodos de autenticação.

Sendo um protocolo utilizado na camada de Enlace, EAP não requer endereçamento IP para seu funcionamento e pode ser utilizado redes de comutação de circuitos, redes via cabo e redes sem-fio (ABOBA, 2004).

De acordo com o padrão IEEE 802.1X, este protocolo também pode ser encapsulado por outros protocolos de camadas superiores como o protocolo TCP e UDP (ABOBA, 2004).

2.6 *Transport Layer Security* (TLS) e *Secure Sockets Layer* (SSL)

São protocolos de criptografia utilizados para trasnportar informações de forma segura criando um túnel entre cliente e servidor.

2.7 PEAP / EAP-MSCHAPv2

Aumenta a seguranca do protocolo EAP por utilizar Transport Layer Security (TLS) para encryptação dos dados.

Com PEAP os frames EAP são transportados por um tunel criptografado por TLS.

3 O PROTOCOLO RADIUS

Como descrito anteriormente o protocolo RADIUS provê serviços de Autenticação, Autorização e Bilhetagem fazendo uso de uma base de dados centralizada.

O processos previstos para um servidor RADIUS são seguem o padrão IEEE 802.1X, citado anteriormente.

O equipamento autenticador, ao detectar um suplicante, envia frames de requisição de identidade EAP a este solicitando sua identificação. Em resposta o suplicante envia um pacote de resposta de identidade EAP contendo contento uma identificação do suplicante, como o nome de usuário.

Ao receber os frames de resposta EAP do suplicante o Autenticador os encapsula em um pacote Radius do tipo *Access-Request* e envia ao Servidor de Autenticação.

Ao receber um pacote de *Access-Request* o Servidor de Autenticação inicia uma negociação com o suplicante para decidir o metodo EAP que será utilizado para a troca de informações de autenticação.

Decidido o metodo de autenticação o Servidor de Autenticação Requisita as informações de autenticação ao Suplicante. Se as credenciais contidas da resposta do suplicente forem encontradas na base de informações do Servidor este responde com uma pacote de *Access-Accept*, caso contrário, responde com uma pacote de *Access-Reject*.Toda a comunicação entre o Servidor de Autenticação e Suplicante é traduzida pelo Autenticador. Ao receber um pacote Radius de *Access-Accept* do Servidor o Autenticador libera o acesso à rede para o suplicante (HASSEL, 2002).

3.1 Formatos de Pacotes

O formato geral dos pacotes RADIUS seguem o modelo conforme o seguinte diagrama:

Header:	TIPO	IDENTIFICADOR	TAMANHO	AUTENTICADOR
Payload:	PARES DE ATRIBUTO E VALOR			

Tabela 1 - Formato do Pacote do Protocolo RADIUS

O Header do pacote tem o total de 20 bytes, sendo: 1 byte para o TIPO, 1 byte para o IDENTIFICADOR, 2 bytes para o TAMANHO e 16 bytes para o AUTENTICADOR.

O campo CODIGO indica o tipo de pacote, que pode ser um destes:

CODIGO	TIPO
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
12	Status-Server (experimental)
13	Status-Client (experimental)
255	Reserved

Tabela 2 - Tipos de Pacotes do Protocolo RADIUS

O campo IDENTIFICADOR é utilizado para associar corretamente as respostas as requisições enviadas.

O campo TAMANHO contém o tamanho de todo o pacote incluindo os campos CODIGO e IDENTIFICADOR assim como o payload.

O campo AUTENTICADOR é utilizado para garantir a integridade do payload. Existem dois tipos de autenticadores: *Request Authenticator* e *Response Authenticator*.

O Request Auhenticator é utilizado em pacotes dos tipos *Authentication-Request* e *Accounting-Request*. Este autenticador é gerado aleatóriamente.

O *Response Auhenticator* é utilizado em pacotes dos tipos *Access-Accept*, *Access-Reject*, e *Access-Challenge*. Este autenticador é gerado através de uma função *hash*:

MD5(TIPO+IDENTIFICDOR+TAMANHO+AUTENTICADOR+AVPs+SECRET)

Onde o sinal de '+' é um operador de concatenação e *SECRET* é o segredo compartilhado (*Shared Secret*) entre o cliente e o servidor RADIUS (RIGNEY, 2000).

3.2 Tipos de PACOTES

3.2.1 *Access-Request*

São utilizados pelo cliente RADIUS ou dispositivo autenticador para requisitar autenticação e serviços ao servidor.

Access-Request tem ao menos dois atributos no seu *payload*: o nome do usuário e seu *password*.

3.2.2 *Access-Accept*

São enviados pelo servidor quando um *Access-Request* é atendido com sucesso. O campo IDENTIFICADOR desse tipo de pacote precisa coincidir com o *Access-Request* atendido.

O *payload* desses pacotes contém AVPs que descrevem os serviços autorizados.

3.2.3 *Access-Reject*

Este tipo de pacote é enviado pelo servidor quando a autenticação não é bem sucedida ou quando os serviços requisitados não foram autorizados.

3.2.4 *Access-Challenge*

Para garantir a autenticidade do usuário é possível enviar pacotes do tipo *Access-Challenge* periodicamente, porém, alguns clientes não suportam esse tipo de processo por isso tratam-no como *Access-Reject*.

3.3 *Shared Secret*

É uma senha que deve ser configurada tanto no servidor como no cliente para aumentar a segurança ou para habilitar acesso ao servidor para clientes especificos.

O *Shared Secret* é utilizados em todos os processos onde há a necessidade de proteger os dados de possíveis interceptações. Como visto anteriormente também é utilizado para a autenticação dos dados de pacotes.

3.4 *Attribute-Value Pairs* (AVP)

São informações referentes aos serviços que podem ser habilitados pelo servidor ou requeridos pelo cliente. Os dados são codificados seguindo o modelo TLV (*Type Length Value*) ou Tipo, Tamanho e Valor; com Tipo ocupando 1 byte, Tamanho pelo menos 4 bytes e Valor tem tamanho variável.

As categorias de AVPs não fazem parte do escopo deste trabalho porque são utilizados em equipamentos mais especializados, comos Servidores de Acesso de Provedores de Serviços de Internet.

O campo tamanho para os atributos comporta-se da mesma forma que o mesmo campo para os pacotes, ou seja representam o tamanho total do bloco TLV. Esse campo precisa ser de no mínimo 4 bytes por que seu tamanho mínimo para acomodar um atributo sera composto por 1 byte para Tipo, 2 bytes para Tamanho e 1 byte para Valor que é o tamanho mínimo de codificação de um atributo (HASSEL, 2002).

3.5 Tipos de Atributos

Os valores dos atributos podem ser codificados conforme a tabela abaixo:

Tipo do Atributo	Tamanho em bytes	Intervalo lógico de Valores	Exemplo
Integer	4	32 bits (sem sinal)	6 / 256
Enumerated	4	32 bits (sem sinal)	1 = Callback-Login 2 = Framed-Compression
String	1-253	Variável	“username”
IP Address	4	32 bits	0xC0AEF324
Date	4	32 bits (sem sinal)	0xC0AEF324
Binary	1	1 bit	0 / 1

Tabela 3 - Codificação de Atributos

Os tipos de atributos não são inseridos os pacotes do protocolo RADIUS. Eles são relacionados em um arquivo de dicionário (*dictionary.conf*) que é carregado pelo servidor RADIUS e traduzidos de acordo com cada Attributo.

4. PROCEDIMENTOS

Para a execução dos procedimentos de configuração utilizou-se uma máquina virtual com a distribuição Linux Ubuntu como servidor Radius e de Banco de Dados MySQL.

Utilizou-se de um roteador *Cisco Linksys RT120N* como cliente Radius configurado com *WPA-Enterprise*.

O DHCP (*Dynamic Host Configuration Protocol*) foi habilidado permitindo que o Roteador fosse configurado automaticamente quando ingressado na rede e para que distribuir **IP** para os demais computadores que venham a se conectar na rede sem fio. Com exceção das configurações exibidas nas imagens abaixo não foram feitas configurações adicionais além das configurações de fabrica.

4.1 Configuração do Roteador

A rede interna padrão do roteador é 192.168.1.0/24 como pode ser visto nas figura, mesma rede em que será configurado o servidor Radius.

Figura 1 - Configuração do Nome da Rede Sem Fio

Atribuiu-se o Nome da Rede (SSID) como 'MinhaRede' como pode ser visto na figura.

Figura 2 - Configuração do Modo de Segurança da Rede Sem Fio

Configuração de um roteador sem fio para autenticação em um servidor RADIUS. Utilizou-se como segredo compartilhado, *shared secret*, o texto "testing123".

4.2 Configuração de Computador para Acesso a Rede Sem Fio

Para testar as configurações utilizou-se um notebook com Windows 7 com placa de rede sem fio. O Windows 7 oferece vários modulos de segurança para conexção sem fio. A rede sem fio foi configurada utilizando as instruções do artigo *Configure 802.1X Wireless Clients Running Windows XP with Group Policy* do site da Microsoft Technet.

Abriu-se o painel de configuração de redes sem fio conforme a figura:
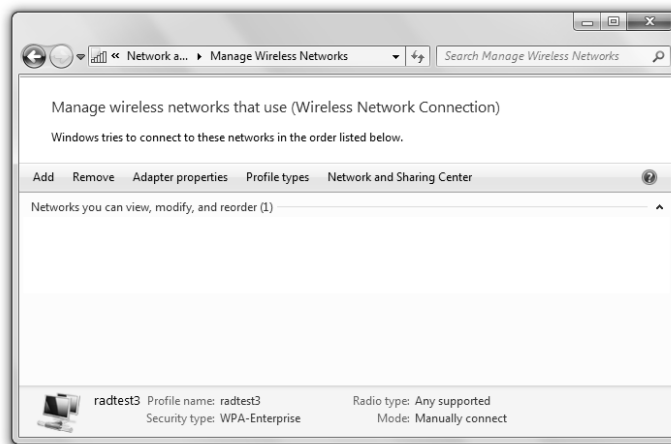
Figura 3 - Gerenciador de Redes Sem Fio do Windows 7

Clicou-se no botão add para adicionar uma nova rede sem fio.
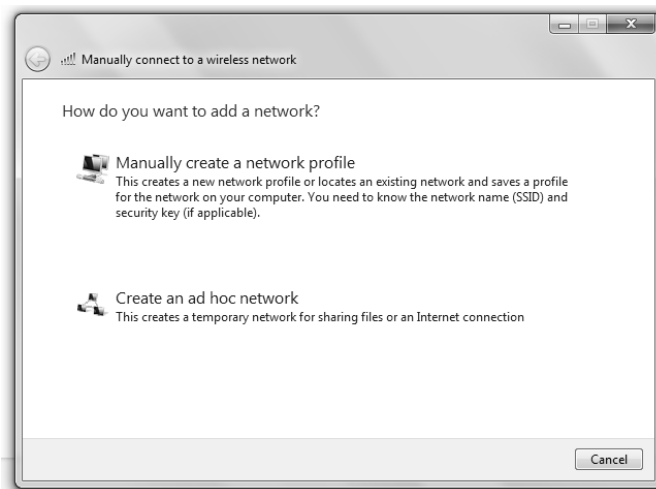
Figura 4 - Configuração Manual do Perfil de Rede

Neste ponto optou-se pela configuração manual de um perfil de rede (*Manually create a network pofile*).

As configurações foram realizadas de acordo com a seguinte imagem.
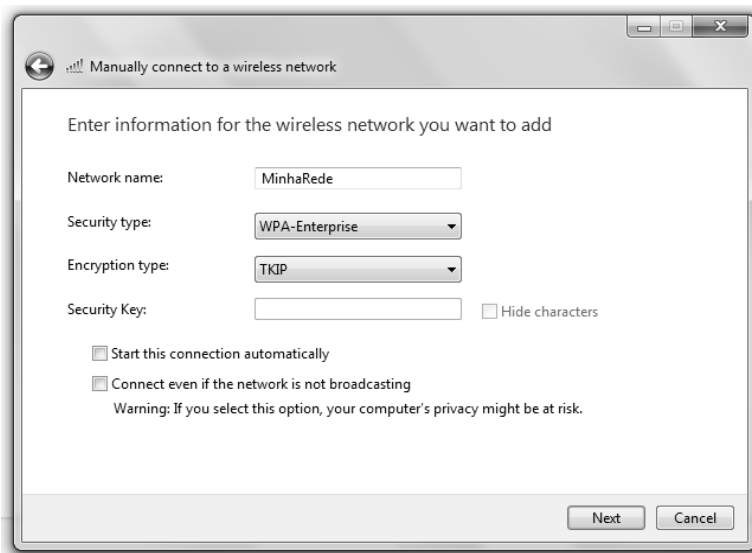
Figura 5 - Configuração do Modo de Segurança do Windows 7

O nome da rede (*network name*) deve ser configurado exatamente como foi configurado no roteador: 'MinhaRede'. Após a conifuração exibida clicou-se em no botao '*Next*'.
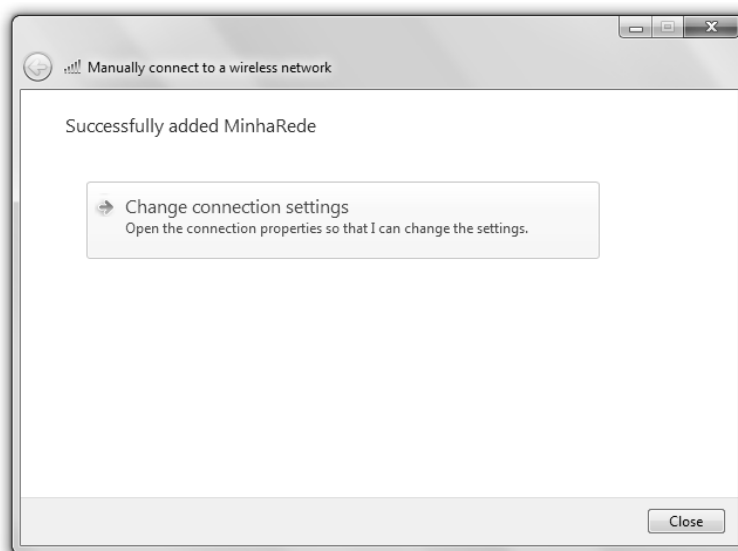
Figura 6 - Alteração os Parâmetros de Configuração

Em seguida clicou-se em '*Change connection settings*' para abrir as configurações avançadas.
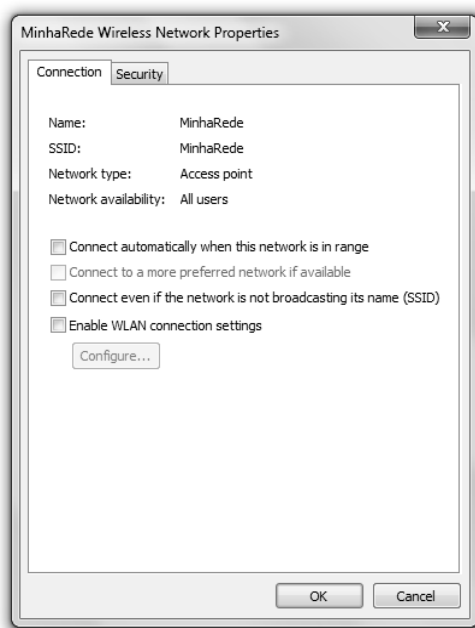
Figura 7 - Configuração dos Parâmetros de Conexão

As caixas de seleção (*CheckBoxes*) foram desmarcadas para que o sistema não tente se conectar automaticamente na rede sem fio. Em seguida clicou-se na aba '*Security*'.
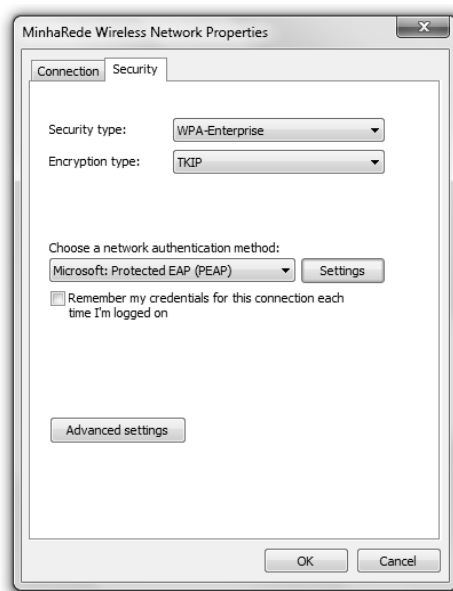
Figura 8 - Configuração dos Parâmetros de Segurança

Clica-se em ‘*Advanced Settings*’ para configurar o modo de autenticação do 802.1X. Deve-se marcar a caixa de seleção ‘*Specify authentication mode*’ e selecionar ‘*User authentication’* no campo correspondente. Essas configurações são as mais comuns e garantem que a autenticação seja realizada em nível de usuário.

Clica-se em no botão ‘OK’ para retornar a caixa de diálogo anterior.
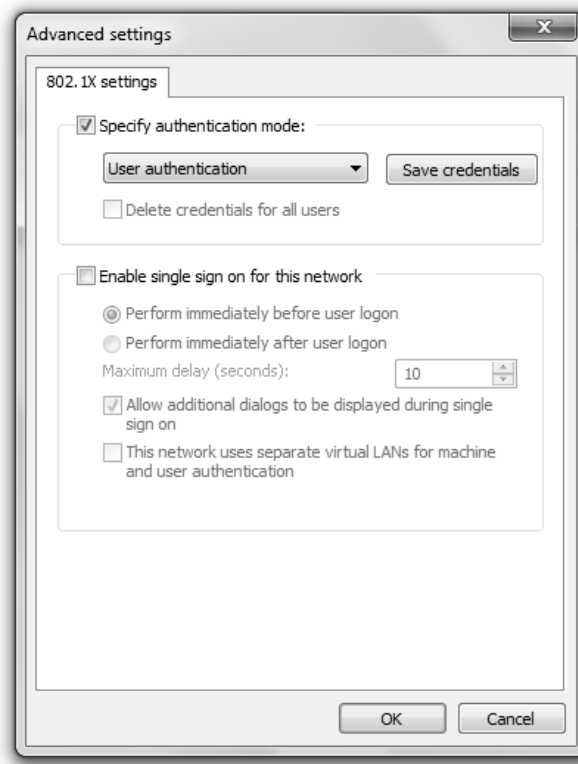
Figura 9 - Configuração do Modo de Autenticação do 802.1X

‘*Security type*’ deve ser aterada para o mesmo tipo de segurança configurada préviamente no roteador, ‘*WPA-Enterprise*’. O tipo de encryptação para ‘TKIP’ e o metodo de autenticação ‘*Microsoft: Protected EAP (PEAP)*’, já mencionÓado na seção REFERENCIAL TEÓRICO. Clica-e em ‘Settings’ para configurações de segurança adicionais.
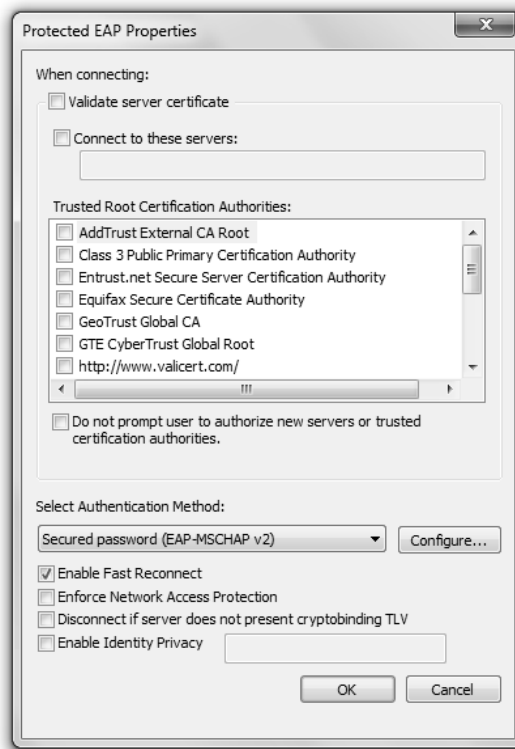
Figura 10 - Configuração Avançada do Modo PEAP

Desabilita-se a Caixa de seleção ‘Validate server certificate’ porque não estão sendo utilizandos certificados digitais. O metodo de autenticação deve ser ‘*Secured password (EAP-MSCHAP v2)*’. Este é um metodo de autenticação EAP deselvolvido pela Microsoft que é suportado pelo freeRADIUS. Ainda é preciso confirurar esse método clicando no botão ‘*Configure...*’ e desmarcando a caixa de seleção confome a figura abaixo:

Figura 11 - Configuração do MSCHAPv2

Finalizadas as configurações basta clicar em ‘OK’ para todas as caixas de diálogo abertas ou ‘Concluir’ quando for o caso.

Isso conclui a configuração do Windows 7 para autenticação no servidor freeRADIUS através do roteador.

O sistema exibirá uma caixa de diálogo solicitando nome de usuários e senha:

Figura 12 - Solicitação dos Dados para Autenticação

Ainda não será possível acessar a rede sem fio porque o servidor freeRadius não configurado. Qualquer tentativa de acesso resultará em falha.

4.3 Instalaçao e Configuração Básica do Servidor freeRADIUS

Para o computador servidor utilizou-se o uma máquna virtual do VirtualBox com a distribuição Linux Ubuntu 8.10.

Optou-se pela instalação do servidor freeRADIUS, por ser gratuito e ser bem documentado em livros.

A instalação é bastante simples. Baste ter a lista de repositórios do Ubuntu atualizada e executar os comandos:

```
apt-cache search freeradius
```

Este commando é utilizado para percorrer a lista de pacotes disponíveis que contém o texto free-radius. Se os repositórios estiverem configurados corretamente e o computador tiver acesso a internet será exibida uma lista dos pacotes disponíveis como no exemplo abaixo:

```
freeradius - a high-performance and highly configurable RADIUS server
freeradius-common - FreeRADIUS common files
freeradius-dbg - debug symbols for the FreeRADIUS packages
freeradius-utils - FreeRADIUS client utilities
libfreeradius-dev - FreeRADIUS shared library development files
libfreeradius2 - FreeRADIUS shared library
freeradius-dialupadmin - set of PHP scripts for administering a FreeRADIUS server
freeradius-iodbc - iODBC module for FreeRADIUS server
freeradius-krb5 - kerberos module for FreeRADIUS server
freeradius-ldap - LDAP module for FreeRADIUS server
freeradius-mysql - MySQL module for FreeRADIUS server
freeradius-postgresql - PostgreSQL module for FreeRADIUS server
```

Observa-se que há muitos pacotes disponíveis para extender as configurações do freeRADIUS. A princípio sera instalado o pacote o primeiro pacote da lista acime ‘freeradius’. Para isso utiliza-se o comando:

```
apt-get install freeradius
```

O commando baixará o pacote correspondente e o instalará em poucos minutos.

Depois de concluída a instalação é preciso configurar o freeRADIUS para que possamos utilizá-lo.

São dois os arquivos necessários para a configuração mais básica do freeRADIUS: *clients.conf* e *users.conf*.

Foram adicionadas as seguintes linhas no arquivo *clients.conf*:

```
client 192.168.1.1
{
secret = testing123
shortname = Roteador1
}
```

Estas linhas adicionam permissão, para o cliente configurado com o ip indicado, autenticar 'na base do freeRADIUS. Obseve os parametros entre as chaves "{}". O parametro '*secret*' é a configuração so shared secret. Como exibido anteriormente o roteador foi configurado com o mesmo segredo.

O parametro 'shortname' é apenas um texo para identificar mais facilmente o roteador podento ser preenchido com qualquer texto.

É necessária a configuração do arquivo 'users.conf' para inserir um usuário para autenticação. Foi inserida a seguinte no início do deste arquivo.

```
user1  User-Password := "user1"
```

Esta linha adiciona o usuário 'user1' a base de autenticão e o atribui a senha 'user1'. Este usuário foi adicionado a base de texto, que é a forma mais simples de se adcionar um usuário para autenticação no freeRADIUS.

Observe que não são adicionados outros parâmetros a configuração, como AVPs. Isso porque esta sendo configurado um ponto de acesso em um roteador

doméstico. Este tipo de roteador não exige configurções adicionais para fornecer acessos a outros serviços.

A mudança nos arquivos de configuração não entrarão em vigor até que o servidor seja reiniciado. Para isso pode-se usar o comando:

```
/etc/init.d/freeradius restart
```

Porém, como estamos trabalhando em um ambiente de testes é interessante observar o comportamento do servidor. Para isso utilizaremos os seguintes comando:

```
/etc/init.d/freeradius stop

freeradius -X
```

A primeira encerra o servidor freeRadius. A segunda linha inicializa o servidor freeRadius em modo ‘*Full Debug*’. Neste modo de operação o servidor exibe o processamento das requisições de autentiação diretamente no console.

Agora é possivel realizar a autenticação através do servidor freeRadius. Para isso, basta inserir os dados de autenticação na caixa de dialogo exibida pelo sistema solicitando informações de autenticação.

É possível acompanhar o progresso da autenticação observando a saída de *Debug* do freeRADIUS.

```
.
.
.
Sending Access-Accept of id 10 to 192.168.1.1 port 32805
	MS-MPPE-Recv-Key = 
0xbebe26001d09f56fd8f4ca749e1b06884c76cbafdac8dac9241be1c980244e82
	MS-MPPE-Send-Key = 
0xb4dd5753d0b0e1b943280b8f10292b3075c33a7e78a333e105ad270adf68d8ca
	EAP-Message = 0x030a0004
	Message-Authenticator = 0x00000000000000000000000000000000
	User-Name = "user1"
Finished request 8.
.
.
.
```

Exemplo de saida de Debug do servidor freeRADIUS em modo ‘*Full Debug*’. A A saída exibe informações de processamento dos pacotes que chegam ao servidor e também indormeções das decisões tomadas de acordo com as configurações e base de autenticação.

4.3 Configuração do Servidor freeRADIUS para Autenticação Utilizando Base de Dados do MySQL

Utilizar a base de dados em texto é bastante fácil, mas não é uma alternativa viável para redes com um número muito grande de usuários. Uma alternativa para esse problema é a utilização de uma base de autenticação gerenciável utilizando MySQL.

Com uma base de autenticação hospedada em um banco de dados é possível criar páginas na intranet para cadastros de novos usuários não havendo a necessidade de abrir os arquivos de configuração toda a vez que for necessário incluir ou excluir um usuário.

A configuração do freeRADIUS para autenticação em base de dados MySQL está concluída, mas ainda é preciso fazer a instalação e configuração do MySQL para que as configurações tenham efeito.

Para instalar o MySQL utilizamos o comando:

```
apt-get install mysql-server
```

Será iniciado o assisente de instalação do myql. É importante informar a senha de administrador do banco de dados. Esta senha será utilizada posteriormente para confiuração de acesso ao banco de dados.

Para facilitar o processo de configuração recomenda-se a instalação do modulo freeradius-mysql, que instalar-a uma coleção de arquivos uteis e pre-configurados para o acesso a base de dados.

Utiliza-se o seguinte comando para instalar ao modulo MySQL para o freeRADIUS.

```
apt-get install mysql-server
```

Serão necessárias algumas alterações nos arquivos de configuração do freeradius para que seja possível consultar a nova base de dados.

No arquivo /etc/freeradius/radiusd.conf é preciso que a seguinte linha esteja descomentada:

```
$INCLUDE sql.conf
```

Os arquivos de configuração do freeRADIUS utilizam o mesmo padrão para comentário de diversas distribuições linux utilizando o caractere '#' como marcador de comentário. Se a linha mencionada estiver precedida do caractere mencionado, basta apagar o caractere.

Quando descomentada, esta linha carrega o arquivo de configuração */etc/freeradius/sql.conf* que contém informações de como o freeradius deve acessar o servidor de banco de dados.

O arquivo *sql.conf* contem uma unica seção identificada por *sql{...}*, onde todas as configurações estão dentro das chaves.

Este arquivo vem configurado, por padrão, para acesso a banco de dados MySQL. Por esse movito não há a necessidade de grandes alterações, bastando apenas informar o endereço ip do servidor de banco de dados, o nome de usuário com a cesso a base de dados e a senha para acesso.

Seguem os campos que precisam ser editados com as configurações realizadas para os testes.

```
server = "localhost"
```

```
login = "root"

password = "admin"

radius_db = "radius"
```

A primeira linha indica o endereõ do servidor de banco dados onde estará a base de dados.

A segunda e terceira linhas indicam o nome do usuário e a senha que deverão ser utilizados pelo freeRADIUS para acessar o banco de dados.

A quarta linha indica qual o banco de dados deverá ser acessado.

Há outras configurações como acessos a tabelas específicas, mas manteremos as configurações padrões para maior simplicidade.

Ainda é preciso indicar ao servidor RADIUS que a tipo de processamento este deverá utilizar o banco de dados MySQL.

Esta configuração é feita no arquivo */etc/init.d/freeradius/sites-enabled/default*, descomentando a entrada '*sql*' na seção *authorize*.

Esta seção indica au servidor freeRADIUS que as requisições de acesso também poderão ser autenticadas utilizando-se da base de dados no MySQL.

O mesmo deve ser feito no arquivo /etc/init.d/freeradius/inner-tunnel para que o freeRADIUS execute o mesmo processo no tunnel TLS criado pelo protocolo PEAP.

Outras seções podem ser habilidadas para configurações mais completas como por exemplo: accounting, que popula tabelas com informações para bilhetagem e; session, utilizada para impedir que o mesmo usuário realize conexões simultaneas na rede, ou seja que um usuário compartilhe sua senha com outro e ambos tenham acesso a rede simultaneamente.

Ainda é preciso criar a base de dados no MySQL e popula-la com informações de autenticação.

A instalação do modulo freeradius-mysql provê um script SQL para a criação o banco de dados e suas tabelas conforme as pre-configurações no arquivo sql.conf.

Esse script encontra-se na pasta */etc/freeradius/sqls/mysql/* no arquivo schema.sql. Para que seja possível importar as informações do script para o MySQL é preciso primeito criar o banco de dados.

O acesso ao MySQL é feito utilizando-se o seguinte comando:

```
mysql -u root -p
```

Onde root é o nome do usuário. Será solicitada a senha de root que foi configurada durante a instalação do mysql.

Uma vez realizado o acesso ao MySQL deverá ser executado o sequinte comando para a criação do banco de dados RADIUS.

```
CREATE DATABAS radius;
```

Será exibita uma mensagem dsemelhante a '*Query OK, 1 row affected (0.00 sec)*' for obtido sucesso;

Com o comando '*quit*' pode-se se desconectar do MySQL para proceder com a configuração.

Utiliza-se seguinte comando para importar criar as tabelas no banco:

```
mysql -u root -p radius < /etc/freeradius/sql/mysql/schema.sql
```

Este commando executa o *script schema.sql* no prompt do mysql, criando assim todas as tabelas necessárias para uso do freeRADIUS.

O scritpt criara ass tabelas de acordo com a figura:

Tables_in_radius
radacct
radcheck
radgroupcheck
radgroupreply
radpostauth
radreply
radusergroup

Figura 13 - Extrutura do Banco de Dados ‘radius’

Para os testes de autenticação precisaremos criar entradas com nomes de usuários e senha de forma semelhante ao que foi realizado no arquivo de configuração user.conf. A tabela utilizada para armazenar informações de autenticação é a tabela *radcheck* que tem o seguinte formato:

Field	Type	Null	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
username	varchar(64)	NO	MUL		
attribute	varchar(64)	NO			
op	char(2)	NO		==	
value	varchar(253)	NO			

Figura 14 - Extrutura da Tabela ‘radcheck’

Para inserir uma entrada de autenticação na tabela radcheck faz-se o acesso ao MySQL, assim como feito anteriormente, e utiliza-se o seguinte comando:

```
insert into radcheck (username, attribute, op, value) values (‘user2’, ‘User-Password’, ‘==‘, ‘user2’);
```

Este commando inserirá uma entrada na tabela correspontente a entrada realizada no arquivo *user.conf*, mas desta vez, para o usuário ‘user2’ com senha ‘user2’;

Basta reiniciar o servidor freeRADIUS para que as configurações de acesso a base de autentiação MySQL tenham efeito. Recomenda-se a utilização do modo *Full Debug* mencionado anteriormente.

O acesso pelo requisitante é realizado da mesma forma inserindo as nome de usuário e senha quando solicitado pelo sistema.

É possível visualizar os logs de autenticação fazendo-se o acesso no MySQL e utilizando os comandos:

```
use radius

select * from radpostauth;
```

5. CONCLUSÃO

Com base nas configurações realizadas em laboratórios e nas dificuldades encontradas para configurar o acesso nos computadores

A princípio as leva-se algum tempo para realizar as instalações e configurações, mas com a prática é possível facilmente configurar um servidor em menos de uma hora para uma pequena rede.

Embora a configuração dos terminais dos usuários seja um pouco complicada exigindo pequeno manual de instruções para que os próprios usuários configurem seus equipamentos para acesso a rede. A utilização do protocolo RADIUS garantirá um maior controle do acesso à rede sem fio e fortificará a política de segurança.

Os acessos ou ingressos rede podem ser registrados em arquivos de log ou em um banco de dados, dependendo da configuração. Dessa forma o administrador da rede pode controlar o uso da rede sem fio ou coletar estatísticas de acessos simultâneos para prover um maior número de ponto de acessos para atender a maiores demandas. O controle estatístico de acesso pode assim garantir a qualidade de trafego na rede sem fio.

Ainda assim existem soluções mais transparentes aos usuários como os pontos de acesso chamado *HotSpot*, que funcionam como um adendo a autenticação rádios ou uma camada extra que torna a configuração por parte do usuário mais amigável ou até imperceptível.

Para acesso a uma rede com *HotSpot* o usuário se conecta ao ponto de acesso que, antes de permitir o ingresso do usuário na rede sem fio solicita a entrada de informações de autenticação, mas comumente o nome usuário e a senha.

A entrada dessas informações são realizas através do navegador de internet por intermédio de uma página que pode ser customizada de acordo com estabelecimento em que o *HotSpot* será instalado.

Hotéis e Aeroportos disponibilizam redes sem fio com *HotSpot* para seus clientes cadastrando estes na base de autenticação.

Estes estabelecimentos podem então cobrar pelo serviço fornecido ou oferecê-los como cortesia a seus clientes mantendo o controle da quantidade de usuários conectados a rede e garantindo assim a qualidade da mesma.

Pontos de acesse *HotSpot* não foram abordados nesse trabalho, mas deve-se mencionar que a base de funcionamento de uma *HotSpot* é baseada no Protocolo RADIUS. Os processos de Autenticação, Autorização e Bilhetagem são executados pelo Servidor Radius no qual devem ser feitas as devidas configurações tal como descritas na seção PROCEDIMENTOS.

Embora este trabalho tenha sua ênfase no protocolo RADIUS, *HotSpot* são uma fonte interessante para trabalhos posteriores.

6. REFERÊNCIAS

ABOBA, B. et allu. *RFC 3748 - Extensible Authentication Protocol*, 2004.

HASSEL, Jonathan. *RADIUS*, O' Reilly: 1 ed, 2002.

RIGNEY, C. et allu, *RFC 2865 - Remote Authentication Dial In User Service (RADIUS)*, 2000.

RIGNEY, C. et allu, *RFC 2866 - RADIUS Accounting*, 2000.

WALT, Dirk van der. FreeRADIUS - Beginner's Guide, Packet Publishing: 1 ed, 2011.

http://www.vivaolinux.com.br/artigo/Freeradius-servidor-radius-eficiente-e-completo/, *Freeradius - servidor radius eficiente e completo*; acesso em 12/11/2011

http://technet.microsoft.com/en-us/library/cc771557%28WS.10%29.aspx, *Configure 802.1X Wireless Clients Running Windows XP with Group Policy*; acesso em 15/11/2011